

## REMARKS/ARGUMENTS

Claims 1-20 are pending in the present application. Claims 3, 13 are canceled; claims 1, 7, 9, 11, 17 and 19 are amended. Reconsideration of the claims is respectfully requested.

Amendments made to the claims 1, 11 and 19 are supported in the description on page 19, lines 15-17, page 21 lines 5-11, page 22 lines 3-4 and page 24 lines 18-30. Amendments to claims 7, 9 and 17 are supported by the respective independent claims on which the instant claims depend for proper antecedent support.

### I. 35 U.S.C. § 102, Anticipation

The examiner has rejected claims 1, 3, 7, 10, 11, 13, 15, 18 and 19 under 35 U.S.C. § 102 as being anticipated by *Ramme*, U.S. Patent Application Publication No. 2003/0093420, Method and System for Retrieving Sharable Information Using a Hierarchically Dependent Directory Structure, May 15, 2003 (hereinafter *Ramme*). The Examiner states:

Regarding claim 1, Ramme teaches a method comprising:

- A) "determining if the web application includes a reference to at least one shared web module that may be incorporated into a plurality of web applications"; and
- B) "identifying a location of the at least one shared web module; and
- C) "logically merging the at least one shared web module with web modules of the web application, if any; to generate a logically merged web application"

With respect to A), the examiner notes that Ramme teaches the claimed features of "determining if the web application includes a reference to at least one shared web module that may be incorporated into a plurality of web applications" as [For information which is sharable among two or more entities such as applications which have module files stored in module directories, the sharable information is stored in one of the module directories, and a link to that directory is stored in each of the directories of the modules that share the information] (see Ramme, paragraph [0013] on page 1).

With respect to B), the examiner notes that Ramme teaches the claimed features of "identifying a location of the at least one shared web module" as [A search may be performed by first searching the directory of the module. If the file is not found, then the link is used to access the directory upon which the module's directory depends, so that it may be searched in a similar manner. The search may continue iteratively from one directory to another, until the file is located] (see Ramme, paragraph [0031] on page 3).

With respect to C), the examiner notes that Ramme teaches the claimed features of "logically merging the at least one shared web module with web modules, of the web application" as [For example, this particular arrangement of the files and links imposes a hierarchical structure on the directory storage, in the

**form of a directed acyclic graph]** (see Ramme, paragraph [0035] together with Figure 1 elements (120,122, 124, 126)).

Office Action dated February 7, 2008, p. 4.

A prior art reference anticipates the claimed invention under 35 U.S.C. § 102 only if every element of a claimed invention is identically shown in that single reference, arranged as they are in the claims. *In re Bond*, 910 F.2d 831, 832, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990). All limitations of the claimed invention must be considered when determining patentability. *In re Lowry*, 32 F.3d 1579, 1582, 32 U.S.P.Q.2d 1031, 1034 (Fed. Cir. 1994). Anticipation focuses on whether a claim reads on the product or process a prior art reference discloses, not on what the reference broadly teaches. *Kalman v. Kimberly-Clark Corp.*, 713 F.2d 760, 218 U.S.P.Q. 781 (Fed. Cir. 1983). In this case, each and every feature of the presently claimed invention is not identically shown in the cited reference, arranged as they are in the claims.

With regard to claim 1, the claim as currently amended is as follows:

A method of generating a logically merged web module for a web application, comprising:

determining if the web application includes a reference to at least one shared web module, that is capable of being incorporated into a plurality of web applications, in a shared web module designation file, wherein the shared web module description file includes all descriptors that reference the shared web modules;

specifying a path to a location of the at least one shared web module; and logically merging the at least one shared web module with web modules of the web application, in accordance with the shared web module designation file, to generate a logically merged web application.

Under the standards of *In re Bond*, *Ramme* does not anticipate claim 1 because *Ramme* does not teach the claimed features of, “... determining if the web application includes a reference to at least one shared web module that is capable of being incorporated into a plurality of web applications; in a shared web module designation file, wherein the shared web module description file includes all descriptors that reference the shared web modules;” “specifying a path to a location of the at least one shared web module;” and “logically merging the at least one shared web module with web modules of web applications in accordance with the shared web module designation file; to generate a logically merged web application.” The Examiner asserts various portions of *Ramme*.

Regarding the feature of, “...determining if the web application includes a reference to at least one shared web module that is capable of being incorporated into a plurality of web applications in a shared web module designation file, wherein the shared web module description file includes all descriptors that reference the shared web modules; ...” the Examiner asserts the following portion of *Ramme*:

“For information which is sharable among two or more entities such as applications which have module files stored in module directories, the sharable information is stored in one of the directories, and a link to that directory is stored in each of the directories of the modules that share the information.”

*Ramme*, paragraph [0013], page 1.

The cited portion of *Ramme* teaches sharable information is stored in one of the directories and linking directory structures to form a chain of directories. *Ramme* teaches storing sharable information in one of the directories, therefore *Ramme* has decided what is sharable and where to place the sharable information. However, *Ramme* does teach to determine if a web application includes a reference; because *Ramme* has already defined the content of the one directory as sharable.

Further the teaching of directories in *Ramme* fails to teach “the shared web module description file includes all descriptors that reference the shared web modules,” of the claimed feature. Thus, *Ramme* fails to disclose the claimed feature. Accordingly, *Ramme* does not anticipate claim1.

With regard to the claimed feature of, “specifying a path to a location of the at least one shared web module;” the Examiner asserts the following portion of *Ramme*:

“...a search may be performed by first searching the directory of the module. If the file is not found, then the link is used to access the directory upon which the module's directory depends, so that it may be searched in a similar manner. The search may continue iteratively from one directory to another, until the file is located.”

*Ramme*, paragraph [0031], page 3.

The cited portion of *Ramme* teaches providing a link enabling a search for a file to traverse directory structures until the file is found. *Ramme* provides a reference to a directory from a directory, as shown in Figure 1 of *Ramme*. *Ramme* thus teaches a search is required to locate a file. However, *Ramme* fails to teach specification of a location of at least one shared module as recited in the claim. *Ramme* instead teaches starting a search in the initial directory of the module and, if the file is not found, traversing to the next directory using the link provided. Because *Ramme* does not have a reference to the location of the desired file, *Ramme* must follow the defined directory chain to search for the file. *Ramme* therefore fails to disclose the claimed feature. Accordingly, *Ramme* does not anticipate claim1.

Regarding the claimed feature of, “...logically merging the at least one shared web module with web modules of the web application in accordance with the shared web module designation file; to generate a logically merged web application ...” the Examiner asserts the following portion of *Ramme*:

“The link may be a real link, such as a Unix link, or it may be implemented as a file with a directory specification stored in association with the directory storage area.

The link file may include all necessary information for accessing another

software component, i.e., may include information on a dependent directory storage area and optionally, may include information on software components available at the dependent directory storage area.”

*Ramme*, paragraph [0057], [0058] page 5.

The cited portion of *Ramme* teaches a logical tree structure for directories. *Ramme* teaches linking directory structures together, rather than merging modules within a web application as recited in the claim. For example, *Ramme* shows in Figure 1 links between directory structures forming a hierarchical structure, rather than merging modules in an application as recited in the claim. In fact, *Ramme* has nothing to do with merging application modules.

In contrast the feature requires the merging “in accordance with the shared web module designation file,” and *Ramme* is silent on the subject matter. Nothing is evident in the teaching of *Ramme* to suggest how files are to be merged as is indicated in the current feature. Therefore *Ramme* fails to disclose the claimed feature. Accordingly, *Ramme* does not anticipate claim1.

*Ramme* fails to teach or suggest the features of claim 1 as amended. Therefore the subject matter is distinguished over the teaching of *Ramme*. Independent claims 11 and 19 have similar subject matter and are therefore also distinguished over the teaching of *Ramme*. Since claims 2, 4-10, and 12, 14-18 and 20 depend from independent claims 1, 11 and 19 respectively, the same distinctions between *Ramme* and the claimed features in the independent claims apply equally well for these dependent claims. Further claims 3 and 13 have been canceled.

Accordingly, under the standard of *In re Bond*, the rejection of claims 1-20 has been overcome. Therefore, the rejection of claims 1, 3, 7, 10, 11, 13, 15, 18 and 19 under 35 U.S.C. § 102 has been overcome.

## **II. 35 U.S.C. § 103, Obviousness (2, 4, 5, 9, 12, 17 and 20)**

The examiner has rejected claims 2, 4, 5, 9, 12, 17 and 20 under 35 U.S.C. § 103 as being obvious over *Ramme* in view of *Spotswood et al.*, US Patent Application Publication No. 200410255293, System and Method For Using A Classloader Hierarchy to Load Software Applications, December 16, 2004 (hereafter *Spotswood*).

With regard to claim 2, the examiner asserts:

Regarding claim 2, most of the limitations of this claim have been noted in the rejection of claim 1. However, *Ramme* does not expressly disclose the claimed feature of "loading the logically merged web application into a web container". *Spotswood et al.*, from the same or similar field of endeavors, discloses the application server constructs the application container with the application components in .the order in which they were retrieved, resulting in a hierarchical classloader structure in the newly constructed

application (see Spotswood, paragraph [0076] on page 5 together with Figure 6). Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references to achieve the claimed feature of "loading the logically merged web application into a web container" (as cited in claim 2). Such combination would have permitted teaching's Spotswood to allow Ramme's to better control over the reloading and namespace separation of individual modules, including EJB's (see Spotswood, paragraph [0019]).

Office Action dated February 7, 2008, pp 8-9.

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). The prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In determining obviousness, the scope and content of the prior art are... determined; differences between the prior art and the claims at issue are... ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or non-obviousness of the subject matter is determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966). "Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue." *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). "*Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.*" Id. (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006))."

Claim 2, as originally provided, states:

The method of claim 1, further comprising:  
loading the logically merged web application into a web container.

The Examiner failed to state a *prima facie* obviousness rejection against claim 2 because the proposed combination does not teach or suggest all the features of claim 2. Specifically, the proposed combination does not teach the features of, "determining if the web application includes a reference to at least one shared web module, that is capable of being incorporated into a plurality of web applications, in a shared web module designation file, wherein the shared web module description file includes all descriptors that reference the shared web modules," "specifying a path to a location of the at least one shared web module," and "logically merging the at least one shared web module with web modules of the

web application, in accordance with the shared web module designation file, to generate a logically merged web application.”

*Spotswood* fails to teach or suggest the claimed features missing from the teaching of *Ramme*. *Spotswood*, as in paragraph [0020], is directed toward, “...allowing individual software modules to be reloaded in memory without forcing other modules to be reloaded at the same time.” Thus, the teachings of *Spotswood* are unrelated to the above-identified features of claim 2.

As shown above, *Ramme* does not teach or suggest the claimed features. Therefore combining the module reloading teaching of *Spotswood* with the teaching of *Ramme* cannot produce the results of the claimed invention.

*Spotswood* teaches away from the instant claims because the addition of *Spotswood* in the proposed combination would defeat the entire purpose of *Ramme*, as well as be contrary to the claimed invention.. *Ramme* is directed toward, “...shared use of modules or software components is enabled through directory links...” while *Spotswood* teaches at paragraph [0054] of page 3, “...isolates applications so that application A cannot see the classloaders or classes of application B.” The two purposes of the cited references are therefore contradictory.

Because *Spotswood* teaches away from claims 2, 4, 5, 9 12, 17 and 20, and from *Ramme*, no motivation exists to combine *Spotswood* and *Ramme*, in accordance with *KSR Int'l.*, as proposed. Accordingly, a *prima facie* obviousness rejection against claims 2, 4, 5, 9 12, 17 and 20 has failed to be stated. Therefore, the rejection of claims 2, 4, 5, 9, 12, 17 and 20 under 35 U.S.C. § 103 has been overcome.

### **III. 35 U.S.C. § 103, Obviousness 6, 8, 14 and 16**

The Examiner rejected claims 6, 8, 14 and 16 under 35 U.S.C. 103(a) as obvious over *Ramme* in view of *Croney et al.*, US Patent Application Publication No. 200410255233 A1, Utilizing Common Layout and Functionality of Multiple Web Pages, December 16, 2004, (hereafter *Croney*). This rejection is respectfully traversed. The Examiner states:

Regarding claim 6, most of the limitations of this claim have been noted in the rejection of claim 1. However, *Ramme* does not expressly disclose the claimed feature of "determining a priority associated with the at least one shared web module" and "resolving any conflicts between shared web modules in the at least one shared web module and conflicts between the at least one shared web module and web modules of the web application, if any". *Croney et al.*, from the same or similar field of endeavors, discloses the server system can be configured to control which of multiple master pages are utilized to form a resulting page based upon predetermined criteria that are satisfied by the occurrence of one or more events or circumstances (see *Croney et al.*, paragraph [0056]). *Croney et al.* further discloses that the modules of the server system can be configured with appropriate computer-executable instructions to recognize predetermined criteria

that imposed by a master page, client system, or server system and to recognize when these predetermined criteria have been satisfied (see *Croney et al.*, paragraph [0058]). Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references to achieve the claimed features of "determining a priority associated with the at least one shared web module" and "resolving any conflicts between shared web modules in the at least one shared web module and conflicts between the at least one shared web module and web modules of the web application, if any" (as cited in claim 6). Such combination would have permitted teaching's *Croney* to allow *Ramme*'s to avoid duplicating of code and content between multiple web pages and unnecessarily filling up the storage with duplicative content (see *Croney*, paragraph [0007]).

Office Action February 7, 2008 pp 11-13.

Claim 6, as originally provided, states:

"The method of claim 1, wherein logically merging the at least one shared web module with web modules of the web application includes:  
determining a priority associated with the at least one shared web module; and  
resolving any conflicts between shared web modules in the at least one shared web module and conflicts between the at least one shared web module and web modules of the web application, if any."

The Examiner failed to state a *prima facie* obviousness rejection against claim 6 because the proposed combination does not teach or suggest all the features of claim 6. Specifically the proposed combination does not teach the features of, "...determining a priority associated with the at least one shared web module;" and "resolving any conflicts between shared web modules in the at least one shared web module and conflicts between the at least one shared web module and web modules of the web application, if any." *Croney* fails to teach or suggest the claimed features missing from the teaching of *Ramme*.

As shown above, *Ramme* does not teach or suggest the claimed features; and *Croney* fails to teach or suggest the claimed features missing from the teaching of *Ramme* therefore, combining the teaching of *Croney* with the teaching of *Ramme* cannot produce the results of the claimed invention. . Accordingly, the proposed combination of references, considered as a whole, does not teach or suggest the claimed features. The Examiner believes otherwise and cites various portions of *Croney*. Regarding claim 6, the Examiner states:

"Regarding claim 6, most of the limitations of this claim have been noted in the rejection of claim 1. However, *Ramme* does not expressly disclose the claimed feature of "determining a priority associated with the at least one shared web module" and resolving any conflicts between shared web modules in the at least one shared web module and conflicts between the at least one shared web module and web modules of the web application, if any". *Croney* et al, from the same or similar field of endeavors, discloses the server system can be configured

to control which of multiple master pages are utilized to form a resulting page based upon predetermined criteria that are satisfied by the occurrence of one or more events or circumstances (see *Croney* et al., paragraph [0056]).

*Croney* et al. further discloses that the modules of the server system can be configured with appropriate computer-executable instructions to recognize predetermined criteria that imposed by a master page, client system, or server system and to recognize when these predetermined criteria have been satisfied (see *Croney* et al., paragraph [0058]). Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references to achieve the claimed features of "determining a priority associated with the at least one shared web module" and "resolving any conflicts between shared web modules in the at least one shared web module and conflicts between the at least one shared web module and web modules of the web application, if any (as cited in claim 6). Such combination would have permitted teaching's *Croney* to allow *Ramme*'s to avoid duplicating of code and content between multiple web pages and unnecessarily filling up the storage with duplicative content (see *Croney*, paragraph [0007]).

Regarding the feature, "determining a priority associated with the at least one shared web module; and resolving any conflicts between shared web modules in the at least one shared web module and conflicts between the at least one shared web module and web modules of the web application, if any" the Examiner believes the following portion of *Croney* teaches this claimed feature:

"[0056] In another embodiment, the server system can be configured to control which of multiple master pages are utilized to form a resulting page based upon predetermined criteria that are satisfied by the occurrence of one or more events or circumstances. This embodiment can be useful, for example, to enable a server system to provide different content/services to different users based upon different subscriptions, privileges, authorization levels, and client system requirements, while at the same time enabling multiple web pages to share a common layout specified by the one or more master pages.

[0058] To enable these alternative embodiments, the modules of the server system can be configured with appropriate computer-executable instructions to recognize predetermined criteria that are imposed by a master page, client system, or server system and to recognize when these predetermined criteria have been satisfied.

[0007] Yet another problem that can be associated with the duplicating of code and content between multiple web pages is that the storage limits of the host system can be filled-up unnecessarily with duplicative content. In other words, a large segment of content or code can undesirably strain the storage capacity of the host when the content segment is redundantly and simultaneously stored for each related web page."

*Croney* paragraphs [0056], [0058] page 5 and [0007] page 1.

*Croney* teaches "control which of multiple master pages are utilized" but does not teach in which order a set of pages may be used. Further the teaching is directed to the provision of different content/services to different users. The teaching of *Croney* therefore teaches away from the common result provided by using shared modules as is taught by *Ramme* and in the claimed features. Further

*Croney* fails to teach resolving a conflict such as between different subscriptions for the same user. *Croney* is silent on the topic of precedence and conflict resolution.

*Croney* fails to teach the claimed features previously shown to be missing from *Ramme*. Therefore a proposed combination of *Croney* and *Ramme* cannot produce the results of the claimed features. The teaching of the cited references, when taken as a whole, therefore cannot provide the results of the claimed features.

The Examiner has thus failed to state a *prima facie* obviousness rejection against claims 6, 8, 14 and 16 using *Ramme* in combination with *Croney*. Therefore, the rejection of claims 6, 8, 14 and 16 under 35 U.S.C. § 103 has been overcome.

#### **IV. Conclusion**

The subject application is patentable over the cited references and should now be in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if, in the opinion of the Examiner, such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: June 30, 2008

Respectfully submitted,

/Theodore D. Fay, III/

Theodore D. Fay, III  
Reg. No. 48,504  
Yee & Associates, P.C.  
P.O. Box 802333  
Dallas, TX 75380  
(972) 385-8777  
Attorney for Applicants

TDF/wr